

Математика Математическое моделирование

Сетевое научное издание

<http://mathmelpub.ru>

Ссылка на статью:

// Математика и математическое моделирование.
2017. № 04. С. 1–17

DOI: 10.24108/mathm.0417.0000067

Представлена в редакцию: 15.07.2017

© НП «НЕИКОН»

УДК 512.5+519.1

Особенности процедуры детерминизации конечных автоматов

Виноградова М.С.¹, Ткачев С.Б.^{1,*},
Кандаурова И.Е.¹

* mathmod@bmstu.ru

¹МГТУ им. Н.Э. Баумана, Москва, Россия

Конечные автоматы широко используются в теории формальных языков, при реализации автоматного подхода к программированию, а также при синтезе алгоритмов логического управления. Для обеспечения однозначности работы алгоритмов синтезированные конечные автоматы должны быть детерминированными. В рамках подхода к синтезу управлений мобильными роботами, например, основанному на применении теории формальных языков, возникают задачи построения различных конечных автоматов, однако такие конечные автоматы, как правило, не будут детерминированными. Алгоритм детерминизации может быть применен к конечным автоматам, заданным различными способами. Наиболее просто основные идеи алгоритма детерминизации можно объяснить, используя представления конечного автомата в виде взвешенного ориентированного графа.

В работе рассматриваются конечные автоматы, представленные как взвешенные ориентированные графы, и подробно разбирается процедура детерминизации конечных автоматов, представленных указанным образом. Приведено подробное изложение алгоритма детерминизации конечных автоматов. Работа алгоритма детерминизации проиллюстрирована большим количеством примеров.

Ключевые слова: конечные автоматы; детерминизация; взвешенные ориентированные графы

Введение

Теория конечных автоматов — один из разделов курса дискретной математики. Конечные автоматы широко используются в теории формальных языков [1,2,3,4], в реализации автоматного подхода к программированию [8], а также при синтезе алгоритмов логического управления, как одиночными, так и многоагентными робототехническими системами [10]. Для обеспечения однозначности работы алгоритма, синтезированный конечный автомат должен быть детерминированным [5,7,9]. Как правило при непосредственном проектировании конечного автомата, как элемента системы логического управления роботами, такой автомат получается детерминированным. Известен подход к синтезу управлений мобильными роботами, основанными на использовании теории формальных языков

[12]. В рамках этого подхода возникают задачи построения конечного автомата для объединения или пересечения языков [10]. Однако результирующий конечный автомат, как правило, не будет детерминированным.

В достаточно общем виде проблема детерминизации конечного автомата рассмотрена в [1,3]. Ниже приведено подробное изложение алгоритма детерминизации конечных автоматов, с использованием их представления, как взвешенных ориентированных графов. Работа алгоритма проиллюстрирована большим количеством примеров. Отметим, что корректность рассматриваемой процедуры для достаточно общего случая обсуждается в [3].

Конечные автоматы и взвешенные ориентированные графы

Конечный автомат представляет собой абстрактную математическую модель дискретного устройства, которое имеет конечное множество возможных состояний Q , среди которых выделяют одно входное состояние (вход), одно или несколько заключительных состояний (выходов). В каждый момент времени устройство может находиться только в одном состоянии из множества Q . Дискретная информация в виде цепочки символов последовательно поступает на вход устройства и посимвольно обрабатывается в режиме "реального времени" [2,8].

Непустое конечное множество символов $V = \{a_1, \dots, a_n\}$, которые могут обрабатываться конечным автоматом, называют входным алфавитом конечного автомата.

Элементы множества V называют буквами или символами.

Буквами или символами могут быть не только собственно буквы какого-либо алфавита, например, латинского, но и дискретные сигналы от датчиков или других устройств. В последнем случае должно быть установлено взаимно однозначное соответствие между поступающими дискретными сигналами и некоторым конечным множеством символов. В англоязычной литературе такие символы называют токенами (token), а процедуру установления требуемого соответствия – токенизацией [6,7].

Из букв или символов формируются входные цепочки или слова. В [1] дано следующее определение слова или цепочки в алфавите V : слово или цепочка в алфавите V – это произвольный кортеж из множества V^k (k -й декартовой степени множества V) для различных $k=0, 1, 2, \dots$ (см. [1], стр. 463).

Можно неформально представить "устройство", моделируемое конечным автоматом, в виде блока управления, считывающей головки автомата и входной ленты. Такое устройство может считывать и обрабатывать цепочку символов, записанных на входной ленте (входную цепочку). Блок управления может в каждый момент времени находиться в одном состоянии из конечного множества Q .

Входная лента представляет собой полубесконечную неограниченную справа, разделенную на отдельные ячейки ленту. В ячейки ленты записываются символы $a_1, a_2, \dots, a_k, \dots$ входного алфавита V , составляющие входную цепочку. Все символы цепочки записыва-

ются последовательно, начиная с первой ячейки без пропусков и занимают ячейки ленты. Например, на рис.1 представлено расположение на ленте цепочки символов *abacab*.



Рис. 1. Принципиальная схема конечного автомата

Считывающая головка автомата всегда позиционируется ровно на одну ячейку входной ленты и после прочтения очередного символа может сдвинуться вправо на одну ячейку или остаться на прежнем месте. Блок управления автомата в каждый момент времени находится только в одном состоянии q из множества Q .

Автомат работает по тактам. На вход конечного автомата в последовательные дискретные моменты времени (такты) $t_1, t_2, \dots, t_k, \dots$ поступают символы входной цепочки (входные сигналы) a_1, a_2, \dots, a_k алфавита V .

Пусть в момент времени t_k считывающая головка автомата установлена напротив k -й ячейки входной ленты, блок управления находится в некотором состоянии $q_i \in Q$. Один такт работы автомата состоит в следующем: автомат считывает содержимое ячейки, и, в зависимости от символа, содержащегося в ячейке, и текущего состояния, блок управления после обработки входной информации переходит в некоторое новое состояние q_j (новое состояние может совпадать с текущим состоянием q_i). Описанием автомата могут быть предусмотрены переходы из одного состояния в другое по пустому слову, при этом автомат не читает ленту и не продвигает головку. Такой такт называют λ -тактом и рассматривается как переход из одного состояния в другое по пустой цепочке. Считают, что переход по пустой цепочке и переход по символу входного алфавита исключают друг друга. Пустую цепочку λ нельзя интерпретировать для конечного автомата как пустое множество, аналог пробела в строке символов или отсутствие символа в ячейке входной ленты.

Если после прочтения последнего символа входной цепочки конечный автомат переходит в одно из заключительных состояний, то говорят, что автомат распознал цепочку.

Может возникнуть такая ситуация, что после прочтения и обработки очередного входного символа считывающая головка остается на прежнем месте и блок управления остается в текущем состоянии, то есть конечный автомат "зависает" в текущем состоянии. В этом случае считают, что автомат не может обработать входную цепочку.

Конечные автоматы можно использовать для формального описания поведения различных систем. Они широко применяются в системах управления, например, для описания сетевых протоколов, работы контроллеров, а также систем логического и событийно-

го управления [7,9,10,11,12]. Конечные автоматы также используются в теории формальных языков для задания процедуры синтаксического анализа текстов, для описания правил работы трансляторов [1,2,3].

Языком конечного автомата называют все множество слов в заданном алфавите V , которое может допустить (распознать) конечный автомат.

Поведение конечного автомата определяется системой его команд. Каждая команда задается записью $q_i a \rightarrow q_j$ или $q_i \lambda \rightarrow q_j$. Первая команда означает, что из состояния q_i по символу $a \in V$ автомат может перейти в состояние q_j , вторая команда означает, что из состояния q_i по пустой цепочке можно перейти в состояние q_j . Возможна ситуация, что $q_i = q_j$, то есть автомат остается в прежнем состоянии. Команда $q_i a \rightarrow q_j$ и команда $q_i \lambda \rightarrow q_j$ являются взаимоисключающими.

Удобным представлением конечного автомата является взвешенный или размеченный ориентированный граф $W = (G; \varphi)$, где $G = (U, E)$ – ориентированный граф, заданный множеством вершин U и множеством дуг E , а φ – весовая функция (или функция разметки), ставящая в соответствие каждой дуге $e_i \in E$ ориентированного графа G некоторый символ входного алфавита или пустое слово λ (метку) [1,2]. Каждому состоянию $q_i \in Q$ блока управления конечного автомата ставится во взаимно однозначное соответствие вершина $u_i \in U$ ориентированного графа, при этом множество дуг и функция разметки определяются системой команд конечного автомата. Если в системе команд автомата есть команда $q_i a \rightarrow q_j$, это означает, что в ориентированном графе из состояния q_i в состояние q_j ведет дуга с меткой a , где a – буква входного алфавита. Если в системе команд есть команда $q_i \lambda \rightarrow q_j$, это означает, что возможен переход из состояния q_i в состояние q_j по дуге с меткой λ . Тогда каждой входной цепочке символов будут соответствовать последовательности вершин $u_0, u_1, \dots, u_n, \dots$, такие, что для каждой пары вершин u_i и u_{i+1} существует дуга из u_i в u_{i+1} для любого i , если существует u_{i+1} (пути) в размеченном ориентированном графе. Среди вершин ориентированного графа выделим вершину, соответствующую начальному состоянию. Вершину ориентированного графа, соответствующую начальному состоянию, в данной работе будем обозначать короткой заходящей стрелкой, а вершины, соответствующие заключительным состояниям будем обозначать короткой исходящей стрелкой (рис. 2).

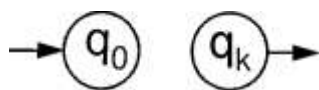


Рис. 2. Графические обозначения начального q_0 и заключительного состояний q_k .

Отметим, что последовательность вершин $u_0, u_1, \dots, u_n, \dots$ в ориентированном графе, представляющем рассматриваемый конечный автомат, может быть конечной или бесконечной. В первом случае ей соответствует конечный путь, а во втором – бесконечный. Наличие бесконечной последовательности вершин (бесконечного пути) означает существование контура в графе. Например, пусть конечный автомат M задан следующим обра-

зом: множество состояний $Q=\{q_1, q_2, q_3\}$; входной алфавит $V=\{a,b,c\}$; система команд: $q_1a \rightarrow q_2$, $q_1b \rightarrow q_2$, $q_2c \rightarrow q_3$; входное состояние q_1 , заключительное состояние q_3 . Граф конечного автомата M представлен на рис. 3.

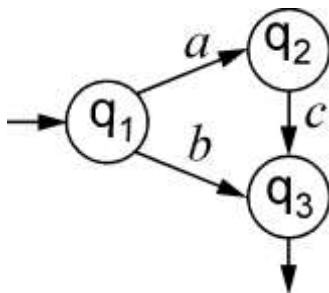


Рис. 3. Граф конечного автомата M

Детерминизация конечных автоматов

При проектировании конечного автомата возможна ситуация, что после прочтения и обработки очередного входного символа a_i из текущего состояния q_i в блоке управления может быть осуществлен переход не в одно, а в несколько состояний (например q_1 и q_m), и необходимо "заранее знать", какое именно состояние следует выбрать, чтобы продолжить чтение входной цепочки. Поведение такого автомата будет неоднозначным, такой конечный автомат называют недетерминированным.

В недетерминированном конечном автомате для каждого символа входной последовательности может существовать несколько (больше одного) состояний, в которые автомат может перейти из текущего состояния. Если существует несколько состояний, в которые автомат может перейти из текущего по одному и тому же символу, то необходимо "заранее знать", какое именно состояние следует выбрать, чтобы входная цепочка была прочитана. Такая ситуация грозит возникновением "конфликтов" и неудобна для реализации. Поэтому возникает задача по исходному конечному автомату получить такой новый автомат, в котором переход из одного состояния в другое был бы однозначным. При этом каждой входной цепочке символов, допускаемой конечным автоматом, должен соответствовать единственный путь в ориентированном графе, представляющем данный автомат. Указанную задачу называют задачей детерминизации конечного автомата.

Следуя [1] зададим конечный автомат M как упорядоченную пятерку, используя функцию переходов δ :

$$M = (V, Q, q_0, F, \delta),$$

где V – входной алфавит; Q – множество состояний; q_0 – начальное состояние; F – множество заключительных состояний; δ – функция переходов.

Функция переходов – это отображение $\delta: Q \times (V \cup \{\lambda\}) \rightarrow 2^Q$, значение функции переходов на упорядоченной паре (состояние $q \in Q$, входной символ из алфавита V или пустая цепочка $\{\lambda\}$) есть множество всех состояний, в которые из данного состояния возможен переход по данному входному символу или пустой цепочке. Это множество, в частности, может быть пустым. Функция переходов может быть задана в виде системы команд: $\delta(q, a) = \{r: qa \rightarrow r\}$. Приведенная в описании функции переходов команда означает,

что из состояния q по символу $a \in V$ можно перейти в множество состояний $\delta(q, a)$, то есть в каждое состояние r из указанного множества ведет дуга с меткой a из состояния q в ориентированном графе, представляющим автомат M .

Определение. Конечный автомат называется детерминированным, если в нем нет команд, содержащих пустую цепочку (нет дуг с меткой λ в ориентированном графе, представляющим этот конечный автомат), и из любого состояния по любому входному символу возможен переход в точности в одно состояние [1], т.е.

$$(\forall q \in Q) (\forall a \in V) (|\delta(q, a)| = 1).$$

Поскольку переход из текущего состояния по фиксированному символу возможен только в одно состояние, то для детерминированного конечного автомата значением функции переходов для любой пары (состояние, входной символ) является одноэлементное подмножество множества состояний. Это подмножество включает в себя ровно одно состояние, в которое осуществляется переход. Это подмножество будет пустым, если из состояния q в ориентированном графе, представляющим автомат M , не выходит дуг, помеченных символом $a \in V$.

Конечные автоматы называются эквивалентными, если они распознают один и тот же язык над алфавитом V [1].

Сформулируем теорему о детерминизации, которая имеет важное значение для решения задачи синтеза конечных автоматов.

Теорема о детерминизации

Для любого конечного автомата может быть построен эквивалентный ему детерминированный конечный автомат [1].

Как было сказано выше, для описания конечного автомата можно использовать ориентированный граф, размеченный над алфавитом V , а также задавать множество состояний Q и функцию переходов, которая может быть задана в виде системы команд. В дальнейшем будем использовать оба подхода, считая их эквивалентными. Выбор подхода будет определяться удобством изложения материала.

Алгоритм построения детерминированного конечного автомата

Построение детерминированного конечного автомата проводится в два этапа.

На первом этапе исходному автомату строится новый конечный автомат, не содержащий команд вида $q_i \lambda \rightarrow q_j$. Можно сформулировать иначе: по графу исходного автомата строится граф нового конечного автомата, не содержащего дуг с меткой λ .

На втором этапе по построенному конечному автомату без λ -переходов строится детерминированный конечный автомат эквивалентный исходному автомату.

Построение конечного автомата без λ -переходов

Пусть исходный конечный автомат задан как упорядоченная пятерка : $M=(V; Q; q_0; F; \delta)$. Построим эквивалентный ему конечный автомат $M_1=(V, Q_1, q_0, F_1, \delta_1)$ без λ -переходов.

Переход от исходного конечного автомата M к эквивалентному автомату M_1 без λ -переходов осуществляется следующим образом.

На первом шаге определяется множество состояний Q_1 нового автомата M_1 . Множество состояний Q_1 включает в себя начальное состояние q_0 исходного автомата M и все состояния из множества состояний Q исходного автомата M , в которые заходит хотя бы одна дуга, помеченная буквой входного алфавита V . Все состояния из множества Q , в которые заходят только дуги с меткой λ , в новое множество состояний Q_1 не включаются.

На втором шаге необходимо определить новое множество дуг конечного автомата M_1 и их меток (функцию переходов автомата M_1).

Все дуги из множества дуг автомата M , метки которых отличны от λ , соединяющие пару состояний из множества Q_1 , включаются в граф автомата M_1 . Все дуги автомата M , имеющие метку дуги λ в множестве дуг конечного автомата M_1 не включаются, а пути в ориентированном графе, представляющим конечный автомат, формируются согласно следующему правилу определения множества дуг автомата M_1 [1]: для любых двух состояний $q_1, q_3 \in Q_1$ $q_1 a \rightarrow q_3$ имеет место тогда и только тогда, когда $a \in V$, в графе M либо существует дуга из q_1 в q_3 , метка которой содержит символ a , либо существует такое состояние q_2 , что из состояния q_1 в состояние q_2 ведет путь ненулевой длины, состоящий из дуг с меткой λ , и из состояния q_2 в состояние q_3 ведет дуга с меткой a .

В множество заключительных состояний F_1 конечного автомата M_1 включаются все состояния $q \in Q_1$, которые являются заключительными состояниями автомата M , или из которых ведет путь ненулевой длины по дугам с меткой λ в одно из заключительных состояний конечного автомата M .

Проиллюстрируем процесс удаления λ -дуг на примерах.

Пример 1. Пусть из состояния q_1 в состояние q_2 ведет путь ненулевой длины, состоящий из дуг с меткой λ , и из состояния q_2 в состояние q_3 ведет дуга с меткой a ($q_1 \xRightarrow{+}_{\lambda} q_2$ и $q_2 \xrightarrow{a} q_3$), в состояние q_2 заходят только дуги с меткой λ (рис. 4).

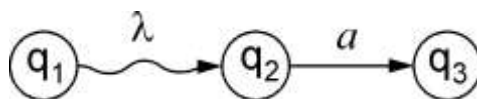


Рис. 4

Результат удаления λ -дуг представлен на рис. 5.

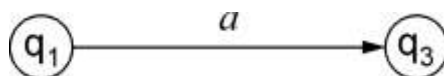


Рис. 5

Пример 2. Пусть из состояния q_1 в состояние q_2 ведет путь ненулевой длины, состоящий из дуг с меткой λ , и из состояния q_2 в состояние q_3 ведет дуга с меткой a ($q_1 \Rightarrow^+_{\lambda} q_2$ и $q_2 \rightarrow_a q_3$), в состояние q_2 заходит кроме дуг с меткой λ , хотя бы одна дуга, помеченная буквой входного алфавита. (рис. 6). На рис.7 показан результат удаления дуг с метками λ .

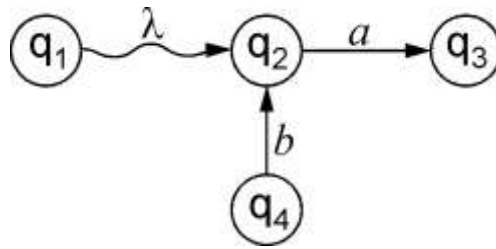


Рис. 6

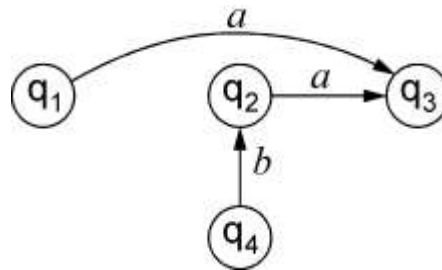


Рис. 7

Пример 3. Пусть из состояния q_1 в состояние q_2 ведет путь не нулевой длины, состоящий из дуг с меткой λ , и из состояния q_2 в состояние q_1 ведет дуга с меткой a ($q_1 \Rightarrow^+_{\lambda} q_2$ и $q_2 \rightarrow_a q_1$) (рис. 8).

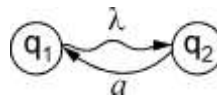


Рис. 8

В результате удаления λ -дуг получим дугу, ведущую из состояния q_1 в тоже состояние (петлю) (рис. 9).

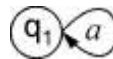


Рис. 9

Пример 4. Пусть из состояния q в состояние f ведет путь не нулевой длины, состоящий из дуг с меткой λ и состояние f является заключительным (рис 10).



Рис. 10

Тогда после удаления λ -дуг состояние q станет заключительным (рис 11).



Рис. 11

Построение детерминированного конечного автомата

Пусть $M_1 = (Q_1; V; q_0; F_1; \delta_1)$ – конечный автомат без λ -переходов.

Построим эквивалентный M_1 детерминированный конечный автомат M_2 , в котором из любого состояния конечного автомата по любому входному символу возможен переход ровно в одно состояние.

Множество состояний нового конечного автомата Q_2 есть некоторое подмножество множества всех подмножеств множества состояний конечного автомата M_1 ($Q_2 \subseteq 2^{Q_1}$).

Каждое отдельное состояние $S = \{q_{s1}, \dots, q_{sk}\}$ конечного автомата M_2 определяет некоторое подмножество множества состояний конечного автомата M_1 : $S \subseteq Q_1$.

Начальным состоянием конечного автомата M_2 является одноэлементное подмножество $\{q_0\}$, содержащее начальное состояние конечного автомата M_1 .

Заключительными состояниями конечного автомата M_2 являются все подмножества Q_1 , которые содержат хотя бы одну заключительную вершину конечного автомата M_1 .

Состояния конечного автомата M_2 будем называть состояниями–множествами. Каждое такое состояние–множество есть отдельное состояние конечного автомата M_2 , но не множество его состояний. Для исходного конечного автомата M_1 это есть именно множество его состояний. Каждое подмножество состояний старого конечного автомата M_1 "свертывается" в одно состояние нового конечного автомата [1].

Функция переходов нового конечного автомата M_2 определяется следующим образом: из состояния–множества S по входному символу a конечного автомата M_2 переходит в состояние–множество, представляющее собой объединение всех множеств состояний конечного автомата M_1 , в которые этот конечный автомат переходит по символу a из каждого состояния множества S

$$\delta_2(S, a) = \bigcup_{q \in S} \delta(q, a)$$

Можно видеть, что конечный автомат M_2 является детерминированным по построению.

Формально новый конечный автомат M_2 определяется следующим образом: $M_2 = (Q_2, V, \{q_0\}, F_2, \delta_2)$, где $Q_2 \subseteq 2^{Q_1}$ множество состояний автомата M_2 , F_2 – множество заключительных состояний конечного автомата M_2 . Множество F_2 состоит из таких состояний–множеств S , что в каждом из них имеется хотя бы одно заключительное состояние автомата M_1 , то есть $F_2 = \{S: S \cap F_1 \neq \emptyset, S \in Q_2\}$. Функция переходов автомата M_2 имеет вид: $(\forall S \subseteq 2^{Q_1})(\forall a \in V) \delta_2(S, a) = \bigcup_{q \in S} \delta(q, a)$.

Среди состояний нового конечного автомата может быть состояние, соответствующее \emptyset , причем $\delta_2(\emptyset, a) = \emptyset$ для любого входного символа a .

Попад в такое состояние, конечный автомат M_2 уже его не покинет. Состояние \emptyset детерминированного конечного автомата M_2 является поглощающим.

Поглощающим состоянием конечного автомата называют такое состояние его q , что для всех входных символов a $\delta(q, a) = q$.

$\delta_2(S, a) = \emptyset$ тогда и только тогда, когда в старом конечном автомате M_1 для каждого состояния $q \in S$ из множества состояний S $\delta_1(q, a) = \emptyset$, т.е. в графе M_1 из каждого такого состояния q не выходит ни одна дуга, помеченная символом a .

Рассмотрим метод детерминизации конечных автоматов, называемый методом "вытягивания".

Алгоритм метода "вытягивания"

В конечном автомате без λ -дуг (пустых дуг) M_1 определяем все множества состояний, достижимых из начального.

1. Для каждого символа входного алфавита a находим множество состояний, достижимых из начального по дугам с метками a $\delta(q_0, a)$. Каждое такое множество в новом автомате является состоянием-множеством, непосредственно достижимым из начального.

2. Для каждого из определенных на предыдущем шаге состояний-множеств S и каждого символа входного алфавита a находим множество $\bigcup_{q \in S} \delta(q, a)$. Все полученные на этом шаге состояния будут состояниями нового детерминированного автомата M_2 , достижимыми из начальной вершины $\{q_0\}$ по пути длины 2.

3. Повторяем шаг 2 до тех пор, пока не перестанут появляться новые состояния-множества (включая пустое).

В результате выполнения шагов 1-3 получим все состояния нового детерминированного автомата M_2 , достижимые из начальной вершины $\{q_0\}$ по пути длины n , $n \geq 2$.

4. Формируем множество заключительных состояний. Состояние-множество нового детерминированного конечного автомата будет заключительным, если оно включает хотя бы одно заключительное состояние автомата M_1 .

Примеры построения детерминированного конечного автомата

Пример 5.

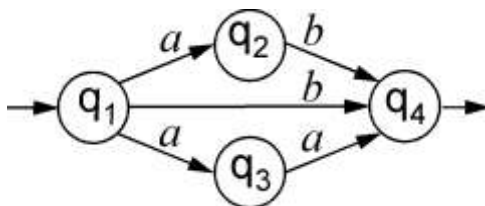


Рис. 12. Конечный автомат M_1

Детерминизируем конечный автомат $M_1 = (Q_1, V, \{q_1\}, F_1, \delta_1)$, представленный на рис. 10, $Q_1 = \{q_1, q_2, q_3, q_4\}$, $V = \{a, b\}$, входное состояние q_1 , множество входных состояний $F_1 = \{q_4\}$, функции переходов: $\delta_1(q_1, a) = \{q_2, q_3\}$, $\delta_1(q_1, b) = \{q_4\}$, $\delta_1(q_2, b) = \{q_4\}$, $\delta_1(q_3, a) = \{q_4\}$.

Конечный автомат M_1 не содержит λ -переходов, поэтому сразу переходим к этапу 2, собственно детерминизации.

Чтобы детерминизировать автомат, воспользуемся методом "вытягивания". Процедуру вытягивания начнем с начального состояния q_1 . Из начального состояния q_1 возможен переход по дуге с меткой a в состояния q_2 и q_3 , получено первое состояние–множество $\{q_2, q_3\}$, по дуге с меткой b в состояние q_4 , получено второе состояние–множество $\{q_4\}$. Рассмотрим состояние–множество $\{q_2, q_3\}$ и найдем функцию переходов из этого состояния. Из состояния $\{q_2\}$ дуги с меткой a нет, поэтому $\delta(q_2, a) = \emptyset$, из состояния $\{q_3\}$ по дуге с меткой a возможен переход в состояние q_4 . В результате получаем $\delta(\{q_2, q_3\}, a) = \{q_4\} \cup \emptyset = \{q_4\}$. Продолжая процедуру вытягивания, получим детерминированный конечный автомат M_2 , все состояния которого достижимы из начальной вершины q_1 . Множество заключительных состояний состоит из единственного состояния $\{q_4\}$.

Для конечного автомата M_2 запишем функцию переходов:

1. $\delta_2(q_1, a) = \{q_2, q_3\}$, $\delta_2(q_1, b) = \{q_4\}$;
2. $\delta_2(\{q_2, q_3\}, a) = \{q_4\} \cup \emptyset = \{q_4\}$, $\delta_2(\{q_2, q_3\}, b) = \{q_4\} \cup \emptyset = \{q_4\}$;
3. $\delta_2(\{q_4\}, b) = \emptyset$, $\delta_2(\{q_4\}, a) = \emptyset$;
4. $\delta_2(\emptyset, b) = \emptyset$, $\delta_2(\emptyset, a) = \emptyset$;

Перечислены все состояния–множества, достижимые из начального, процедура "вытягивания" на этом заканчивается, и в результате получен детерминированный конечный автомат M_2 , граф которого представленный на рис.13.

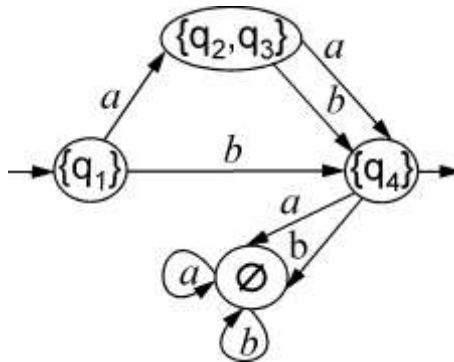


Рис.13. Конечный автомат M_2

Пример 6.

Рассмотрим пример детерминизации конечного автомата M с λ -дугами. Конечный автомат M задан как упорядоченная пятерка $M = (Q, V, q_1, F, \delta)$, где $Q = \{q_1, q_2, q_3, q_4, q_5, q_6\}$ – множество состояний конечного автомата, $V = \{a, b\}$ – входной алфавит, q_1 – входное состояние, $F = \{q_5, q_6\}$ – множество входных состояний, функции переходов:

$\delta(q_1, \lambda) = \{q_2, q_3\}$, $\delta(q_2, \lambda) = \{q_5\}$, $\delta(q_2, a) = \{q_4\}$, $\delta(q_2, b) = \{q_6\}$, $\delta(q_3, a) = \{q_2, q_6\}$, $\delta(q_3, b) = \{q_4\}$, $\delta(q_4, b) = \{q_5\}$, $\delta(q_5, \lambda) = \{q_3\}$, $\delta(q_6, b) = \{q_5\}$.

Граф конечного автомата М представлен на рис. 14.

На рисунке все λ -дуги выделены красным цветом.

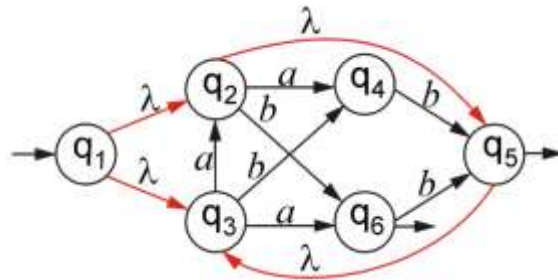


Рис. 14. Конечный автомат М

Построим конечный автомат M_1 без λ -переходов, эквивалентный конечному автомату М.

Множество состояний нового конечного автомата M_1 без λ -переходов $Q_1 = \{q_1, q_2, q_4, q_5, q_6\}$. Состояние q_3 не вошло в множество состояний нового автомата M_1 , так как в него заходят только дуги с метками λ в конечном автомате М, и в новом конечном автомате M_1 это состояние не достижимо от входа. На рис. 15 это состояние показано светло-серым цветом.

Определим новое множество дуг графа конечного автомата M_1 и их меток (функцию переходов автомата M_1). В множество дуг автомата M_1 войдут все дуги графа автомата М, соединяющие пару состояний из нового множества Q_1 автомата M_1 , метки которых отличны от λ . Для того, чтобы не "потерять" цепочки, которые могут быть прочитаны на путях с λ -переходами, рассмотрим эти пути.

Из состояния q_1 по дуге с меткой λ можно перейти в состояние q_2 , а из состояния q_2 по дуге с меткой a – в состояние q_4 , следовательно, из состояния q_1 можно перейти в состояние q_4 по дуге с меткой a , $q_1 \rightarrow_a q_4$. Рассмотрим путь $q_1 \rightarrow_\lambda q_2 \rightarrow_\lambda q_5 \rightarrow_\lambda q_3 \rightarrow_a q_2$. Согласно правилу удаления λ -переходов получим путь $q_1 \rightarrow_a q_2$. Так как в состоянии q_2 заходит не только дуги с метками λ , это состояние остается в множестве состояний Q_1 . Такой переход можно было бы получить, используя путь $q_1 \rightarrow_\lambda q_3 \rightarrow_a q_2$. Из состояния q_2 по дуге с меткой λ можно перейти в состояние q_5 , а из состояния q_5 по дуге с меткой λ – в состояние q_3 и из состояния q_3 по дуге с меткой a – в состояние q_2 , следовательно, получаем путь $q_2 \rightarrow_a q_2$ или петлю. Рассмотрим пути $q_2 \rightarrow_\lambda q_5 \rightarrow_\lambda q_3 \rightarrow_b q_4$ и $q_2 \rightarrow_\lambda q_5 \rightarrow_\lambda q_3 \rightarrow_b q_6$. После удаления λ -переходов получим пути $q_2 \rightarrow_b q_4$ и $q_2 \rightarrow_a q_6$. Из состояния q_5 по дуге с меткой λ можно перейти в состояние q_3 и из состояния q_3 – в состояния q_2 и q_6 по дугам с метками a и в состояние q_4 – по дуге с меткой b . Следовательно, получаем пути $q_5 \rightarrow_a q_2$, $q_5 \rightarrow_b q_4$ и $q_5 \rightarrow_a q_6$.

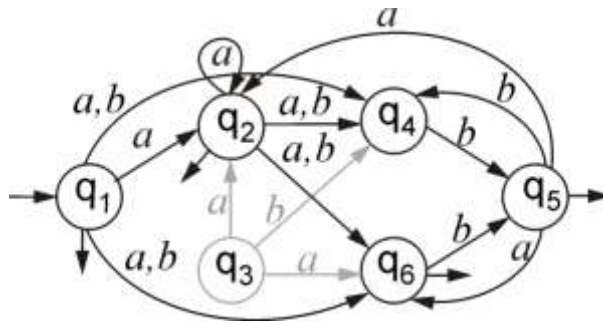


Рис. 15. Конечный автомат M_1

Из состояний q_1 и q_2 по дуге с меткой λ можно перейти в заключительное состояние q_5 , и, следовательно, они войдут в множество заключительных состояний F_1 , $F_1 = \{q_1, q_2, q_5, q_6\}$.

На рис. 15 представлен конечный автомат $M_1 = (Q_1, V, q_1, F_1, \delta_1)$ без λ -переходов.

Чтобы детерминизировать полученный автомат, воспользуемся методом "вытягивания". На рис. 16 изображен конечный автомат M_1 , где для удобства дальнейшей работы все дуги с меткой a сделаем красными, а дуги с меткой b – синими, состояние q_3 не показано на рис. 16, как недостижимое от входного состояния q_1 .

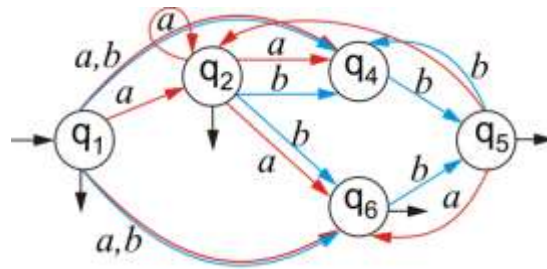


Рис. 16. Конечный автомат M_1

Процедуру вытягивания начнем с начального состояния q_1 , из которого возможен переход по дуге с меткой a в состояния q_2 , q_4 и q_6 , по дуге с меткой b в состояния q_4 и q_6 . Получены новые состояния–множества $\{q_2, q_4, q_6\}$ и $\{q_4, q_6\}$. Из состояния–множества $\{q_2, q_4, q_6\}$ по дуге с меткой a возможен переход в состояния q_2 , q_4 и q_6 , то есть в тоже состояние, а по дуге с меткой b в состояния q_4 , q_5 и q_6 . Применяем процедуру "вытягивания" до тех пор, пока не перестанут появляться новые состояния–множества. Функции переходов конечного автомата M_2 имеют вид:

- 1) $\delta_1(\{q_1\}, a) = \{q_2, q_4, q_6\}$, $\delta_1(\{q_1\}, b) = \{q_4, q_6\}$;
- 2) $\delta_1(\{q_2, q_4, q_6\}, a) = \delta(q_2, a) \cup \delta(q_4, a) \cup \delta(q_6, a) = \{q_2, q_4, q_6\} \cup \emptyset = \{q_2, q_4, q_6\}$;
 $\delta_1(\{q_2, q_4, q_6\}, b) = \{q_4, q_5, q_6\}$;
- 3) $\delta_1(\{q_4, q_6\}, a) = \emptyset$, $\delta_1(\{q_4, q_6\}, b) = \{q_5\}$;
- 4) $\delta_1(\{q_4, q_5, q_6\}, a) = \{q_2, q_6\}$; $\delta_1(\{q_4, q_5, q_6\}, b) = \{q_4, q_5\}$;
- 5) $\delta_1(\{q_5\}, a) = \{q_2, q_6\}$; $\delta_1(\{q_5\}, b) = \{q_4\}$;
- 6) $\delta_1(\{q_2, q_6\}, a) = \{q_2, q_4, q_6\}$; $\delta_1(\{q_2, q_6\}, b) = \{q_4, q_5, q_6\}$;

$$7) \delta_1(\{q_4, q_5\}, a) = \{q_2, q_6\}; \delta_1(\{q_4, q_5\}, b) = \{q_4, q_5\};$$

$$8) \delta_1(\{q_4\}, a) = \emptyset; \delta_1(\{q_4\}, b) = \{q_5\};$$

$$9) \delta_1(\emptyset, a) = \emptyset; \delta_1(\emptyset, b) = \emptyset;$$

В множество заключительных состояний конечного автомата M_2 войдут следующие состояния-множества: $\{q_1\}$, $\{q_2, q_4, q_6\}$, $\{q_4, q_6\}$, $\{q_4, q_5, q_6\}$, $\{q_5\}$, $\{q_2, q_6\}$, $\{q_4, q_5\}$, так как в каждое из этих состояний-множеств входят состояния конечного автомата M_1 , принадлежащие множеству F_1 , $F_1 = \{q_1, q_2, q_5, q_6\}$.

По полученным функциям перехода построим граф детерминированного конечного автомата (рис. 17).

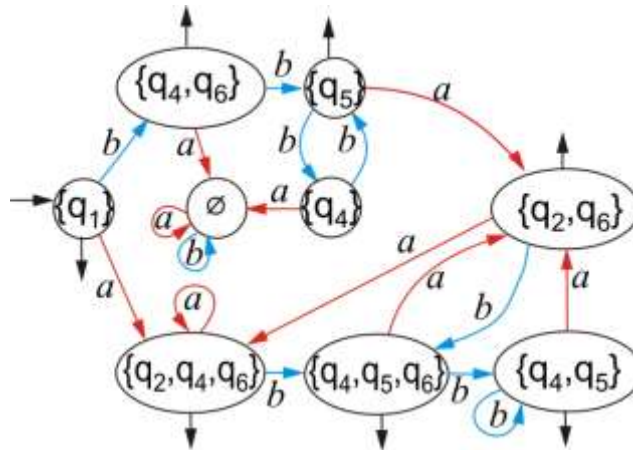


Рис. 17. Конечный автомат M_2

Заключение

Рассмотрен алгоритм детерминизации конечного автомата по графу исходного автомата. Заметим, что предложенный алгоритм можно модифицировать таким образом, чтобы по конечному автомату, заданному формально, получить формальное описание детерминированного конечного автомата, эквивалентного исходному. Использовать представление в виде размеченного ориентированного графа в этом случае не требуется.

Список литературы

1. Белоусов А.И., Ткачев С.Б. Дискретная математика: учеб. для вузов / Под ред. В.С. Зарубина и А.П. Крищенко. 5-е изд. М.: Изд-во МГТУ им. Н.Э. Баумана, 2015. 743 с.
2. Хопкрофт Дж., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений. 2-е изд. М.: Вильямс, 2002. 527 с. [Hopcroft J.E., Motwani R., Ullman J.D. Introduction to automata theory, languages and computation. 2nd ed. Boston: Addison-Wesley, 2001. 521 p.].

3. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. В 2 т. Т.1. М.: Мир, 1978. 612 с. [Aho A.V., Ullman J.D. The theory of parsing, translation and compiling. Vol. 1. Englewood Cliffs: Prentice-Hall, 1972].
4. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979. 536 с. [Aho A.V., Hopcroft J.E., Ullman J.D. The design and analysis of computer algorithms. Reading: Addison-Wesley, 1974. 470 p.].
5. Новорусский В.В. Конечнoавтоматные системы управления (принципы построения и анализ поведения). Новосиб.: Наука, 1982. 269 с.
6. Dantam N., Stilman M. The motion grammar: Analysis of a linguistic method for robot control // IEEE Transactions on Robotics. 2013. Vol. 29. No. 3. Pp. 704-718.
DOI: [10.1109/TRO.2013.2239553](https://doi.org/10.1109/TRO.2013.2239553)
7. Frazzoli E., Dahleh M.A, Feron E. Maneuver-based motion planning for nonlinear systems with symmetries // IEEE Transactions on Robotics. 2005. Vol. 21. No. 6. Pp. 1077-1091.
DOI: [10.1109/TRO.2005.852260](https://doi.org/10.1109/TRO.2005.852260)
8. Шалыто А.А. Автоматное проектирование программ. Алгоритмизация и программирование задач логического управления // Известия РАН. Теория и системы управления. 2000. № 6. С. 63-81.
9. Шалыто А.А. Алгоритмизация и программирование для систем логического управления и "реактивных" систем // Автоматика и телемеханика. 2001. № 1. С. 3-39.
10. Максимов А.А. Один подход к построению конечнoавтоматной управляющей сети // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2012. Спец. вып. 6: Робототехнические системы. С. 14-29.
11. Kobele G.M., Riggle J., Brooks R., Friedlander D., Taylor C., Stabler E. Induction of prototypes in a robotic setting using local search MDL // 9th Intern. Symp. on artificial life and robotics: AROB'2004 (Beppu, Oita, Japan, January 28-30, 2004): Proc. 2004. Pp. 482-485.
12. Горячкин Б.С. Развитие теории конечных автоматов и ее приложений в МГТУ им. Н.Э. Баумана // Инженерный вестник. МГТУ им. Н.Э. Баумана. Электрон. журн. 2015. № 4. С. 538-542. Режим доступа: <http://engbul.bmstu.ru/doc/730377.html> (дата обращения 19.08.2017).
13. Brockett R.W. Formal languages for motion description and map making // Robotics. Providence: Amer. Math. Soc., 1990. Pp. 181–193.

The Determining Finite Automata Process

M.S. Vinogradova¹, S.B. Tkachev^{1,*},

I.T. Kandaurova¹

[*mathmod@bmstu.ru](mailto:mathmod@bmstu.ru)

¹Bauman Moscow State Technical University, Moscow, Russia

Keywords: finite state automata; determinization; weighted directed graph

The theory of formal languages widely uses finite state automata both in implementation of automata-based approach to programming, and in synthesis of logical control algorithms.

To ensure unambiguous operation of the algorithms, the synthesized finite state automata must be deterministic. Within the approach to the synthesis of the mobile robot controls, for example, based on the theory of formal languages, there are problems concerning the construction of various finite automata, but such finite automata, as a rule, will not be deterministic. The algorithm of determinization can be applied to the finite automata, as specified, in various ways. The basic ideas of the algorithm of determinization can be most simply explained using the representations of a finite automaton in the form of a weighted directed graph.

The paper deals with finite automata represented as weighted directed graphs, and discusses in detail the procedure for determining the finite automata represented in this way. Gives a detailed description of the algorithm for determining finite automata. A large number of examples illustrate a capability of the determinization algorithm.

References

1. Belousov A.I., Tkachev S.B. *Diskretnaia matematika* [Discrete mathematics]: a textbook / Ed. by V.S. Zarubin, A.P. Krishchenko. 5th ed. Moscow: Bauman MSTU Publ., 2015. 743 p. (in Russian).
2. Hopcroft J.E., Motwani R., Ullman J.D. *Introduction to automata theory, languages and computation*. 2nd ed. Boston: Addison-Wesley, 2001. 521 p. (Russ. ed.: Hopcroft J.E., Motwani R., Ullman J.D. *Vvedenie v teoriyu avtomatov, iazykov i vychislenij*. 2nd ed. Moscow: Williams Publ., 2002. 527 p.).
3. Aho A.V., Ullman J.D. *The theory of parsing, translation and compiling*. Vol. 1. Englewood Cliffs: Prentice-Hall, 1972. (Russ. ed.: Aho A.V., Ullman J.D. *Teoriia sintaksicheskogo analiza, perevoda i kompiliatsii*. Vol. 1. Moscow: Mir Publ., 1978. 612 p.).

4. Aho A.V., Hopcroft J.E., Ullman J.D. *The design and analysis of computer algorithms*. Reading: Addison-Wesley, 1974. 470 p. (Russ. ed.: Aho A.V., Hopcroft J.E., Ullman J.D. *Postroenie i analiz vychislitel'nykh algoritmov*. Moscow: Mir Publ., 1979. 536 p.).
5. Novorusskij V.V. *Konechnoavtomatnye sistemy upravleniia (printsipy postroeniia i analiz povedeniia)* [Finite-automatic control systems (principles of construction and behavior analysis)]. Novosibirsk: Nauka Publ., 1982. 269 p. (in Russian).
6. Dantam N., Stilman M. The motion grammar: Analysis of a linguistic method for robot control. *IEEE Transactions on Robotics*, 2013, vol. 29, no. 3, pp. 704-718.
DOI: [10.1109/TRO.2013.2239553](https://doi.org/10.1109/TRO.2013.2239553)
7. Frazzoli E., Dahleh M.A., Feron E. Maneuver-based motion planning for nonlinear systems with symmetries. *IEEE Transactions on Robotics*, 2005, vol. 21, no. 6, pp. 1077-1091.
DOI: [10.1109/TRO.2005.852260](https://doi.org/10.1109/TRO.2005.852260)
8. Shalyto A.A. Software automation design: Algorithmization and programming of problems of logical control. *J. of Computer and Systems Sciences Intern.*, 2000, vol. 39, no. 6, pp. 899-916.
9. Shalyto A.A. Logic control and “reactive” systems: Algorithmization and programming. *Automation and Remote Control*, 2001, vol. 62, no. 1, pp. 1-29.
10. Maksimov A.A. One approach to constructing a finite automaton control network. *Vestnik MGTU im. N. E. Bauman. Ser. Priboroostroenie, spec. iss. 6: Robototekhnicheskie sistemy* [Herald of the Bauman MSTU. Instrument Engineering. Spec. iss. 6: Robototechnical systems], 2012, pp. 14-29 (in Russian).
11. Kobele G.M., Riggle J., Brooks R., Friedlander D., Taylor C., Stabler E. Induction of prototypes in a robotic setting using local search MDL. *9th Intern. Symp. on artificial life and robotics: AROB'2004* (Beppu, Oita Japan, January 28-30, 2004): Proc. 2004. Pp. 482-485.
12. Goriachkin B.S. The development of the theory of finite automata and its applications in the Bauman MSTU. *Inzhenernyj vestnik. Bauman MGTU* [Engineering Bulletin], 2015, no. 4, pp. 538-542. Available at: <http://engbul.bmstu.ru/doc/730377.html>, accessed 19.08.2017 (in Russian).
13. Brockett R.W. Formal languages for motion description and map making. *Robotics*. Providence: Amer. Math. Soc., 1990. Pp. 181–193.